

Bibliographic data: JP 2002533805 (A)

METHOD AND APPARATUS FOR EXECUTING PL/SQL STORED CODE OBJECTS IN A DATABASE

Publication date: 2002-10-08

Inventor(s):

Applicant(s):

Classification: - international: G06F12/00; G06F17/30; G06F9/44; (IPC1-7): G06F12/00
- European: G06F17/30S4F9P; G06F9/44G2

Application number: JP20000590047T 19991215

Priority number(s): US19980219253 19981222; WO1999US29921 19991215

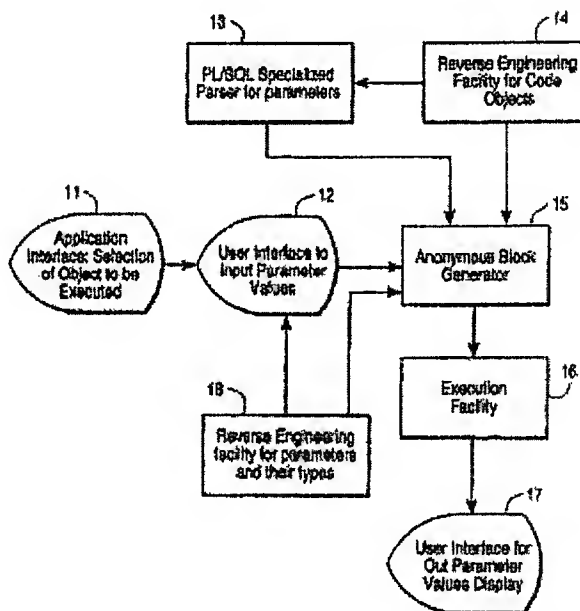
Also published as:

- WO 0038055 (A1)
- ZA 200105302 (A)
- US 7047249 (B1)
- US 6108659 (A)
- IL 143823 (A)
- more

Abstract not available for JP 2002533805 (A)

Abstract of corresponding document:
WO 0038055 (A1)

A system, method and database development tool are disclosed for automatically generating parameters for use in executing a stored code object in a data base, by parsing source codes (Fig. 1) derived from existing database (Fig. 1, 13) which represents stored code object, wherein the parameters may represent complex data types (Fig. 1, 17).



Last updated:
26.04.2011 Worldwide
Database 5.7.23.1; 93p

(19) 日本国特許庁 (J P)

(12) 公表特許公報 (A)

(11) 特許出願公表番号
特表2002-533805
(P2002-533805A)

(43) 公表日 平成14年10月8日 (2002.10.8)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード* (参考)
G 0 6 F 12/00	5 1 3	G 0 6 F 12/00	5 1 3 D 5 B 0 8 2

審査請求 未請求 予備審査請求 有 (全 29 頁)

(21) 出願番号 特願2000-590047(P2000-590047)
(86) (22) 出願日 平成11年12月15日 (1999.12.15)
(85) 翻訳文提出日 平成13年6月22日 (2001.6.22)
(86) 国際出願番号 PCT/US99/29921
(87) 国際公開番号 WO00/38055
(87) 国際公開日 平成12年6月29日 (2000.6.29)
(31) 優先権主張番号 09/219, 253
(32) 優先日 平成10年12月22日 (1998.12.22)
(33) 優先権主張国 米国 (US)

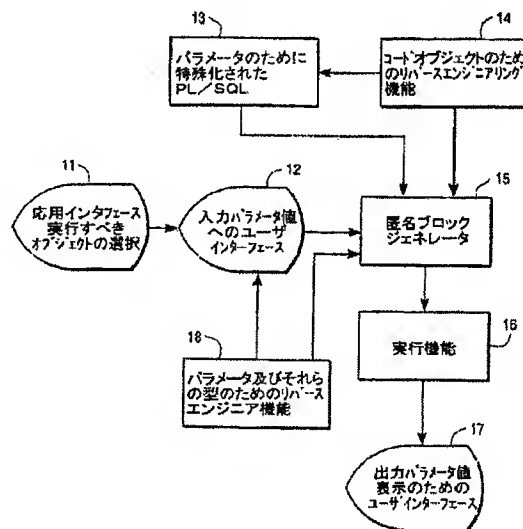
(71) 出願人 コンピュータ アソシエイツ シンク, インコーポレイテッド
アメリカ合衆国・ニューヨーク州
11749・イスランディア・ワン コンピュータ アソシエイツ プラザ
(72) 発明者 ヴィンセント ジョン ケイ
アメリカ合衆国 カリフォルニア州
94804 リッチモンド ショアライン コート 200
(74) 代理人 弁理士 中村 稔 (外9名)
Fターム(参考) 5B082 6A08

最終頁に続く

(54) 【発明の名称】 データベース内に PL / SQL 格納されたコードオブジェクトを実行する方法及び装置

(57) 【要約】

格納されているコードオブジェクトを表すソースコード (図1) を既存のデータベースから導出してパースする (図1の13) ことによって、データベース内に格納されているコードオブジェクトを実行するために使用されるパラメータを自動的に生成し、これらのパラメータが複素データ型を表す (図1の17) ことができるようにしたシステム、方法、及びデータベース開発ツールが開示されている。



【特許請求の範囲】

【請求項1】 少なくともプロセッサ、メモリ、ディスプレイ、入力／出力デバイス、及びデータベースサーバーに接続するための接続性ソフトウェアを有するコンピュータを使用して、コードを自動的に生成する方法であって、

既存コードから導出され、データベース内に存在しているソースコードをパーズして、複素データ型を含むことができるパラメータを生成するステップと、

上記生成されたパラメータを使用して、格納されているコードオブジェクトを実行するのに使用されるコードを作成するステップと、
を含むことを特徴とする方法。

【請求項2】 コンピュータ可読媒体上に埋込まれているデータベース開発ツールであって、

既存コードから導出され、データベース内に存在しているソースコードをパーズして、複素データ型を含むことができるパラメータを生成するために上記媒体上に記録されている手段と、

上記生成されたパラメータを使用して、格納されているコードオブジェクトを実行するのに使用されるコードを作成するために上記媒体上に記録されている手段と、

を備えていることを特徴とするデータベース開発ツール。

【請求項3】 データベース内に存在する格納されたコードオブジェクトを実行するために使用される、複素データ型を含むことができるパラメータを生成するシステムであって、

ディジタルコンピュータと、

既存コードから導出され、データベース内に存在しているソースコードをパーズして、複素データ型を含むことができるパラメータを生成するように構成されているパーズングデバイスと、

上記生成されたパラメータを使用して、格納されているコードオブジェクトを実行するのに使用するためのコードを作成するように構成されている実行デバイスと、

を備えていることを特徴とするシステム。

【請求項4】 少なくともプロセッサ、メモリ、ディスプレイ、入力／出力デバイス、及びオラクル（登録商標）サーバーに接続するための接続性ソフトウェアを有するコンピュータを使用して、オラクルデータベース内のサブプログラムを実行するために使用することができる実行可能な匿名ブロックを生成する方法であって、

- a) オラクルカタログからサブプログラムを選択するステップと、
 - b) 上記サブプログラムをリバースエンジニアすることによって上記サブプログラムのためのソースコードを入手するステップと、
 - c) 上記ソースコードをパースして、上記ソースコードからパラメータの宣言を抽出するステップと、
 - d) 上記パラメータの宣言によって指示されるパラメータ値を使用して、実行可能な匿名ブロックを生成するステップと、
- を含むことを特徴とする方法。

【請求項5】 上記実行可能な匿名ブロックは、P L / S Q L 試験ツールの一部としてオラクルデータベース内のサブプログラムを実行するために使用できることを特徴とする請求項4に記載の方法。

【請求項6】 上記実行可能な匿名ブロックは、P L / S Q L コードカバレージツールの一部としてオラクルデータベース内のサブプログラムを実行するために使用できることを特徴とする請求項4に記載の方法。

【請求項7】 上記実行可能な匿名ブロックは、P L / S Q L プロファイリングツールの一部としてオラクルデータベース内のサブプログラムを実行するために使用できることを特徴とする請求項4に記載の方法。

【請求項8】 上記実行可能な匿名ブロックは、P L / S Q L デバッガーの一部としてオラクルデータベース内のサブプログラムを実行するために使用できることを特徴とする請求項4に記載の方法。

【請求項9】 少なくともプロセッサ、メモリ、ディスプレイ、入力／出力デバイスを有するコンピュータを使用して、オラクルデータベース内のサブプログラムを実行する方法であって、

- a) オラクルカタログからサブプログラムを選択するステップと、

- b) 上記サブプログラムから抽出したデータによって指示されるパラメータ値を使用して、実行可能な匿名ブロックを生成するステップと、
 - c) 上記匿名ブロックを使用して、上記サブプログラムを実行するステップと、
 - d) 上記サブプログラムの実行が完了した後に、OUTパラメータ値を出力するプログラムメカニズムを使用するステップと、
- を含むことを特徴とする方法。

【請求項10】 上記サブプログラムからパラメータ情報を抽出することによって構成されたパラメータリストからOUTパラメータを入手することによってOUTパラメータ値を出力するためのステートメントを構成するステップを更に含むことを特徴とする請求項9に記載の方法。

【請求項11】 少なくともプロセッサ、メモリ、ディスプレイ、入力／出力デバイス、及びオラクルサーバーに接続するための接続性ソフトウェアを有するコンピュータを使用して、オラクルデータベース内のオラクルサブプログラムの有効宣言ステートメントを生成する方法であって、

- a) オラクルデータベースサーバーに対するPL／SQLコードを開発するために使用するように構成されたデータベース開発ツールを準備するステップと、
 - b) オラクルカタログからサブプログラムのソースコードをリバースエンジニアするステップと、
 - c) パラメータ宣言のために上記ソースコードをパースし、次いで上記宣言を変更してそれを有効変数宣言ステートメントにするステップと、
 - d) 上記変数宣言ステートメントを使用して、上記PL／SQLコードを開発するステップと、
- を含むことを特徴とする方法。

【請求項12】 データベースプログラムを開発するシステムであって、

- a) 少なくともプロセッサ、メモリ、ディスプレイ、入力／出力デバイス、及びオラクルサーバーに接続するための接続性ソフトウェアを有するコンピュータと、
- b) 既存コードから導出され、オラクルデータベース内に存在しているソース

コードをパースすることによって、有効P L / S Q Lコードを自動的に生成する
プログラムコードメカニズムと、
を備えていることを特徴とするシステム。

【発明の詳細な説明】

【0001】

(技術分野)

本発明は、コンピュータ関連システム及び方法に関する。より特定的には、本発明は、データベース内のコードオブジェクトの実行に使用するためのパラメトリックデータを自動的に生成するためのシステム及び方法に関する。

【0002】

(従来技術)

データベース内に格納されているコードオブジェクトを実行するためのコードを生成するための効率的な、そして費用有効な手段を提供することが望まれている。これらの格納されているコードオブジェクトは、オブジェクトの試験のために、オブジェクトのコードカバレッジを決定するために、オブジェクトをデバッグするために、その他の類似の理由から、実行しなければならない。殆どのデータベースシステム内のオブジェクトの試験、プロファイリング、デバッグ等のために使用されるコードを生成するには、それが時間を消費すること、及び高度に誤りを犯し易いことから高度に高価であり、非効率的である。例えば、PL／SQL™問合せを使用するオラクル(Oracle™)データベースでは、格納されたコードオブジェクトまたはサブプログラムは2つの方法、即ち

1) サブプログラムを呼出す匿名ブロックを実行する能力を可能にする他に、パラメータのブラインドを処理するオラクル呼出しインタフェース(OCI)のような応用プログラミングインタフェースを使用する、及び

2) サブプログラムを呼出す匿名ブロックを構成するPL／SQL開発ツールを使用する、
の一方で実行することができる。

【0003】

サブプログラムオブジェクトを実行するプログラムを書込むために応用プログラミングインタフェースを使用すると極めて時間がかかり、サブプログラミングを実行する実代的な代替にはならない。サブプログラムを呼出す匿名ブロックを構成することも、極めて時間を消費し得る。SQL*Plus™は、ユーザにサブプ

プログラムを実行させることを可能にする‘実行’コマンドを供給する。パラメータを結合させるために、SQL*Plusは、コマンド‘変数’を供給する。これら2つのコマンドを組合わせて使用することにより、PL/SQLデベロッパーはサブプログラムを実行することができる。しかしながら、SQL*Plus実行機能は、以下の問題の故に、その使用は極めて制限されたものとなっている。(SQL*Plus及びオラクル及びオラクル7 (ORACLE7™) は、ORACLE Corporationの商標である。)

1) ‘実行’コマンドはPL/SQLレコード及びレコードのPL/SQLテーブル内に複素パラメータ型を有しているコードオブジェクトを実行するために使用することができない。オラクル7応用プログラミングインタフェースは、複素パラメータ型の結合を支援せず、従ってパラメータの結合に基づくどのような実施も、それが複素パラメータ型を支援することができないという事実によって制限される。

2) 多数のパラメータを有するコードオブジェクトを実行することは、極めて時間を消費する莫大な‘変数’コマンドの書込みを含む。

【0004】

従来から、データベース試験及びデバッグを自動化するために多くの試みがなされてきた。このような試みの1つは、米国特許第5,361,351号“System & Method for supporting run-time data type identification of objects within a computer program”に開示されているように、変更されたデバッグ情報を格納し、オブジェクトの実際のデータ型を決定するためにデバッグテーブルを構成するコンパイラを使用している。別の試みは、米国特許第5,519,859号“Method & Apparatus for Automatic table selection and generation of structured query language instructions”であって、この特許は、所定のデータテーブル内のデータ及び自動的に選択された出力テーブルから、構造化問合せ言語ステートメントを自動的に生成するシステムを開示している。これらの両方の場合に、法外な量の時間及び努力がデータテーブル及びコンパイラにより開発されるテーブルを準備し、開発するために振り向けられている。他の多くの米国特許がデータベースのためのさまざまなデバッグ及び試験システムを開示しているが、出

願人の知るところでは、それらの何れも、このようなサブプログラムを実行するために必要なパラメータを自動的に生成する本発明の方法及びシステムを提供してはいない。

【0005】

プログラマがパラメータの値を入力できるようにし、サブプログラムを実行した後に出力パラメータの値を表示できるサブプログラムを実行する方法を提供すると有利である。自動的に匿名ブロックを生成して実行する方法は、コーディングの誤りを排除しながら匿名ブロック内のパラメータデータ型を宣言し、同時に宣言されたパラメータのための値を初期化すべきである。またこの方法は、パラメータとして、複雑なユーザが定義した型を取るサブプログラム（手動で行うには極めて困難であり、誤りを犯し易いタスク）を実行できるようにすべきである。開発中にコードオブジェクトの試験を通して、完全な実行機能を容易にすべきである。これ自体は、後刻クライアントプログラムを通して生産環境内でこれらのコードオブジェクトを展開し、実行する時に、論理的な誤りを減少させることによって金銭を節約すべきである。

【0006】

本発明は、コードオブジェクトを実行するために使用することができる完全な実行機能を提供するために、フレームワークを通してこれらの問題に対処する。本発明は、パラメータ全体が結合するのを回避することによって、実行可能な匿名ブロックを自動的に生成する。代わりに、パラメータ型に対応する変数が宣言され、匿名ブロックにおいて初期化される。コードオブジェクトを呼出した後の、参照によって結合されたパラメータに対応する変数は、サブプログラムによって戻される値を含む。変数宣言及び初期化は、ソースコードのパーズング、カタログ内のサブプログラムに関して格納されているパラメータ情報の探索、及びパラメータ値を入手するためのウィザードの使用の組合わせを通して達成される。

【0007】

（発明の概要）

既存のデータベースコードから導出され、格納されているコードオブジェクトを表すソースコードをパーズすることによって、データベース内に格納されてい

るコードオブジェクトを実行するのに使用するためのパラメータ（これらのパラメータは、複素データ型を表すことができる）を自動的に生成するシステム、方法、及びデータベース開発ツールを開示する。

【0008】

また、オラクルデータベース内のサブプログラムを実行するための実行可能な匿名ブロックを生成する方法を開示する。更に、実行されるサブプログラムから抽出されたパラメータ値をそれら自体が使用する匿名ブロックを使用して、オラクルデータベース内のサブプログラムを実行する方法、及びオラクルデータベース内のサブプログラムの有効な宣言ステートメントを生成する方法も開示する。また、既存コードから導出され、データベース内に存在しているソースコードをパースすることによってPL／SQLを自動的に生成するために、コンピュータシステム及びプログラムコードメカニズムからなるデータベースプログラムを開発するためのシステムも開示する。

【0009】

当分野に精通していれば、以下の詳細な説明から本発明の他の実施の形態が明白になるであろう。図示し、説明する本発明の実施の形態は、本発明を遂行するために現時点では最良のモードを例示しているに過ぎない。本発明は、他の、及び異なる実施の形態が可能であり、それらの若干は例示の目的から説明することがあり、また説明の内容は本発明の思想及び範囲から逸脱することなくさまざまに変更することができる。

【0010】

（実施の形態）

本発明は、データベース内のサブプログラムを実行するための方法及び装置を提供する。サブプログラムを実行する能力は、デベロッパの基礎的な要求である。これによって、コードオブジェクトの開発中に、それらの論理を検証することが可能になる。

【0011】

本発明を理解し易くするために、例示の目的のための以下の説明では特定のデータ及び構成を記述する。現在では好ましい実施の形態において、本発明をオラ

クル統合開発環境（IDE）に関して説明する。しかしながら、当分野に精通していれば、本発明は以下の特定の詳細通りではなく、オラクル、Sybase、DB2、Informix等のようないろいろなデータベースシステムで実現できることは理解されよう。本発明を不要に解り難くすることを避けるために、他の例では公知のシステム及びプロトコルを図式的に、またはブロック図で示し、説明する。

【0012】

付加的な従来技術の情報

PL／SQLは、オラクル7サーバーにおいて使用可能なプログラミング言語である。これは、標準SQL言語の手続きの延長である。SQLは、関係型データベースに対する照会を書くために設計された標準問合せ言語である構造的問合せ言語を表している。オラクル7データベース内に実現されているPL／SQLは、実施規則及びデータベースサーバ内の他のビジネス論理に適合させた手続き言語である。手続きプログラミング言語においては、プログラムのための論理は典型的にはサブプログラムとして実現される。PL／SQLでは、サブプログラムは以下の1つの形状を取る。即ち、

1) PL／SQL格納された手順。これは、入力パラメータを取り、出力パラメータを戻すことができるルーチンである。

2) PL／SQL格納された機能。これは、入力パラメータを取り、出力パラメータを戻すことができるルーチンである。更に、格納された機能は、表現式内に使用することができ、またスペック内に宣言されている特定の型の値を戻す。

3) PL／SQL格納されたパッケージ。これは、単一の命名されたエンティティとして識別される格納された機能及び手順の集まりである。

4) PL／SQL格納されたトリガ。これは、データ操作ステートメントがデータベースサーバ内で実行される時に自動的に実行されるルーチンである。

5) 匿名PL／SQLブロック。これは、サーバー上で実行することができる無名セットのPL／SQLステートメントである。

【0013】

一旦展開すると、命名されたルーチンはデータベースサーバ内に存在し、実行することができる。一方、匿名ブロック直ちに実行され、データベースサーバ

一内に格納されない。

【0014】

サブプログラムのためのパラメータは、そのサブプログラムの実行中に読取って変更できるある型のデータベースを含むことができる命名された記憶位置である。パラメータを読取るのか、または変更するのかを支配する規則は、パラメータモードを通してオラクル内に指定されている。パラメータモード‘IN’はパラメータを読取ることではできるが、変更はしないことを指定し、‘OUT’はパラメータを読取ることではできるが、変更はできないことを指定し、‘IN OUT’はパラメータを読取りも、変更もできることを指定する。各パラメータは、そのパラメータのための値の可能範囲、それらの値に対して遂行できる動作、及びそれらの値を格納する方法を指定する。

【0015】

PL／SQLは、標準セットのスカラ型データを支援する。更に、PL／SQLは1組のユーザ（が）定義（した）データ型をも支援する。ユーザ定義型は、特定のデータベーステーブルのデータ定義に基づくことも、またはユーザ定義型定義に基づくこともできる。ユーザ定義型定義は、基本スカラデータ型の合成であることができ、また制限されない数のネスティングを有することもできる。

【0016】

サブプログラムの開発中に、入力パラメータのための異なるパラメータ値が与えられた時に、機能がどのように挙動するかを見るために、PL／SQLプログラムがオブジェクトを実行できるようにすることが重要である。論理が正しくても、無効セットのパラメータが渡されるとオブジェクトは不正確に挙動することがあり得る。開発中のコーディングの誤りを検出するためには、異なるパラメータ値に基づくサブプログラムの実行が必要である。

【0017】

サブプログラムを実行するために、呼出し者は、パラメータ及び関連データ型の全てを知る必要がある。この知識は、

- 1) 匿名ブロックが手順を実行するために構成されているか、または
- 2) サブプログラムを実行するプログラムがオラクル応用プログラマイインタフ

ェースを使用して書かれているか、
には無関係に必要である。

【0018】

サブプログラムを実行するためにこれらの何れかの方法を使用することは、時間を消費するタスクである。前者の方法は、

1) パラメータのための正しいデータ型宣言を有する匿名ブロックを構成し、そして

2) 1組の有効パラメータ値を用いて手動でパラメータを初期化する、
という時間を消費する、そして誤りを犯し易い2つのタスクを行うことを含む。

【0019】

後者の方法は、サブプログラムを正しく実行するために、新しいプログラムのデバッグ及び試験を含む。

【0020】

動作環境

本発明は、一般的に図6に示す要素の若干または全てを含むことができる汎用コンピュータユニット内において、または該ユニットの一部として動作する。汎用システム201は、入力／出力(“I/O”)区分205、1つ以上の中央処理ユニット(“CPU”)207、及びメモリ区分209(それに関連付けられたフラッシュメモリカード211を有する)を有するマザーボード203を含んでいる。I/O区分205は、キーボード226、他の類似汎用コンピュータユニット225、215、ディスク記憶ユニット223、及びCD-ROMドライブユニット217に接続されている。CD-ROMドライブユニット217は、典型的にはプログラム221及び他のデータを含むCD-ROM媒体219を読み取ることができる。以下に詳細に説明するように、これらのプログラムされたコンピュータの論理回路または他の成分は、コンピュータプログラムによって指令される特別に識別された一連の動作を遂行する。

【0021】

好ましい実施の形態

本発明は、[オラクル]データベース内のサブプログラムを実行するための方

法及び装置を提供する。好ましい実施の形態では、本方法は、入力パラメータの値を取入れ、コードオブジェクトを実行し、そして出力パラメータの値を表示する。選択されたコードオブジェクト、選択されたコードオブジェクトのパラメータ値を入力することを可能にするユーザインタフェース、及び出力パラメータの値を表示するためのユーザインタフェースが、プロセス内に使用される。選択されたコードオブジェクトは、そのオブジェクトに対応するソースコードをリバースエンジニアするために必要な情報を供給する。次いでソースコードは、ソースコードから複素パラメータ宣言を抽出する特殊化されたパーザーへ渡される。パラメータの名前、型、及び記憶長に関する情報は全て、オラクルカタログからリバースエンジニアリングされる。カタログからの、及びソースコードからパースされた宣言ステートメントからのパラメータ情報自体は、変数宣言ステートメント及び変数初期化ステートメントを構成するために使用される。カタログからのパラメータ情報は、スカラデータ型のための宣言ステートメントを構成するのに十分である。しかしながら、複素データ型の場合には、実際の宣言は二重にする必要がある。このプロセスにより、複素型に関して完全に実現されないプロセスであるパラメータと値との明示結合をバイパスすることができる。匿名ブロック生成は、サブプログラムを実行するために実際の呼出しを行うストリングを構成することによって完成される。出力変数値を表示させるために、サブプログラムに対する呼出しの後に、DBMS_出力オラクルシステムパッケージに基づく1組の印刷ルーチンが付加される。次いで、生成された匿名ブロックが実行され、DBMS_出力は出力パラメータ値の表示をメッセージする。

【0022】

匿名ブロックは、サーバー上で実行することができるPL／SQLコードの無名ブロックである。命名されたブロックが2つの別個の動作でコンパイルされて実行されるのに対して、匿名ブロックは単一の動作でコンパイルされて実行される。サブプログラムをPL／SQLで実行するためには、そのサブプログラムを呼出す匿名ブロックを構成しなければならない。サブプログラムに対する呼出しは、パラメータ値を渡すことを含む。パラメータは、ある値と関連付けることができる命名された変数である。次いで、サブプログラムは、これらのパラメータ

をソースコード内の変数としてアクセスすることができる。各パラメータは、パラメータ名、パラメータ型、及びパラメータモードを有している。パラメータ名は、そのパラメータに与えられる識別子である。パラメータ型は、あるパラメータが取り得る値の範囲を指定するデータのセットを定義する。パラメータ型は、スカラ、または複素数の何れかであることができる。スカラパラメータ型の例は、**NUMBER**及び**VARCHAR2**を含む。複素パラメータ型の例は、**RECORDS**及び**PL / SQL**テーブルの**RECORDS**を含む。**RECORD**は、スカラ型の合成である**PL / SQL**データ型である。**RECORDS**は、他の**RECORDS**を含む**RECORDS**を有することができるという意味で、ネストすることができる。パラメータモード**IN**、**OUT**、及び**IN OUT**は、どのサブプログラムとも共に使用することができる。**IN**パラメータにより、呼出されているサブプログラムに値を渡すことができる。サブプログラムの内部では、**IN**パラメータは定数として動作し、従って値を割当ててすることはできない。**OUT**パラメータにより、値を呼出者へ戻すことができる。サブプログラムの内部では、**OUT**パラメータは未初期化変数として動作し、従って読取ることはできない。**IN OUT**パラメータにより、値をサブプログラムへ渡し、変更された値を戻すことができる。**IN OUT**パラメータは初期化済みの変数として動作し、従って読取ることも、変更することもできる。

【0023】

図1は、好ましい実施の形態における本発明のブロック図である。ユーザインタフェース成分11は、実行すべきサブプログラムを入手する。ユーザインタフェース成分12は、入力されたパラメータ値を入手する。成分18は、パラメータ名、型、モード等を入手する問合せを発行するモジュールである。次いで、このモジュールはパラメータのリストをメモリ内に構成する。このリストは、ユーザインタフェース成分12を構成するために使用される。パラメータ値情報を集めた後に、成分15は、成分13及び14からの出力を使用して匿名ブロックを構成する。

【0024】

サブプログラムを実行する匿名ブロックを構成するために、好ましい実施の形態において本発明は、以下の入力を受信する。

- 1) データベース内のサブプログラム名及び位置 (図1の成分11)、及び
- 2) 選択されたサブプログラムのためのパラメータ値 (図1の成分12)。

【0025】

本発明は、選択されたサブプログラムに基づいてパラメータ値を入力するためのテンプレートを生成し、個々のパラメータの値を入力するように促す。これを行うべく、図1の成分18はパラメータ型及び各パラメータのサブタイプを識別するために、データベースカタログに問合せる。以下のオラクルカタログに対するオラクル問合せは、格納されている手順のこの情報を与える。格納されている機能、格納されているパッケージ手順、または格納されているパッケージ機能のためのこの情報を抽出するために、類似の問合せを書くことができる。

```
SELECT u.name, o.name, a.argument, a.sequenc#, a.overload#, a.position,
a.type,
DECODE(DECODE(a.type, 29, 3, 69, 11, a.type), 0, 'No Parameters',
1, 'VARCHAR2', 2, 'number', 3, 'NATIVE_INTEGER',
8, 'LONG', 11, 'ROWID', 12, 'DATE', 23, 'RAW',
24, 'LONG RAW', 96, 'CHAR', 106, 'MLSLABEL', 105,
MSLABEL',
250, 'RECORD', 251, 'TABLE', 252, 'BOOLEAN'),
DECODE) NVL(a.default#, 0), 1, 'YES', 'NO') DEFAULT#,
NVL(a.in_out, 0),
NVL(a.level#, 0),
NVL(a.length, 0),
NVL(a.precision, 0),
NVL(a.scale, 0),
NVL(a.radix, 0), decode(o.status, 1, 1, o),
Bitand(b.options, 4)
FROM sys.argument$a, sys.obj$o, sys.user$u, sys.procedure$b
WHERE a.obj#(+) = o.obj#
and o.obj# = b.obj#
```



```

AND o.owner#=b.user#
And o.linkname is null
and o.type=7
and
(
o.owner# in(uid,1/*PUBLIC*/)
or
(o.obj# in(select obj# from sys.objauth$
      where grantee# in (select kzsrorol from ptsql_auth)
      and privilege#=12)/*EXECUTE*/)
)
or
(exists (select null from sys.v_$enabledprivs
      where priv_number in (-144/*EXECUTE ANY PROCEDURE
      */,-142,-143,
      -141/*CREATE ANY PROCEDURE*/)
)
)
)
)

```

【0026】

好ましい実施の形態では、本発明は、パラメータ及びその関連する型のリストをメモリ内に構成する。このリストは、成分が全てスカラである複素パラメータ型のために別々にネストされる。ネストされる複素パラメータ型の場合には、このリスト内には複数のネスティングが存在する。このリストは、入力パラメータ値を入手するためのテンプレートを生成するために使用される。このメカニズムによって、これらの値（これらの値は、パラメータの型によって指定された値の範囲に適合していなければならない）を入力しながら、誤りを検出することができる。

【0027】

図2乃至5は、匿名ブロックを生成するために使用される諸ステップを示すフローチャートである。成分21及び22は、メモリ内におけるパラメータのリストの構成を示している。これらのモジュールからの出力として得られるリストは、パラメータを入力する（図1の成分12）ためにユーザインタフェース成分によって使用され、また匿名ブロック生成成分によって使用される。

【0028】

次いで、本発明は、第1のアイテムから始めてリストを走査する（図2の成分23）。図2の成分24は、そのアイテムが複素パラメータであるか否かを調べる。もしそれが複素パラメータであれば、図3のフローチャートへ進み、複素パラメータを宣言し、初期化する匿名ブロック構成ステップを辿る。そうでなければ、図4のフローチャートへ進み、スカラパラメータを宣言し、初期化するステップを辿る。図2の成分23から始まるステップはパラメータ毎に繰り返され、サブプログラムの全てのパラメータを宣言し、初期化するストリングが構成される。パラメータの宣言、及び初期化が完了した後に、図5のフローチャートは、匿名ブロックの構成を完了させるために使用される本発明の諸ステップを示している。匿名ブロックが構成されると、それはサーバー上で実行され、出力パラメータが出力される。

【0029】

本発明は、匿名ブロックの宣言及び初期化区分を構成するために2つのストリングを保持している。複素パラメータを宣言するために、本発明は先ずオラクルサーバーからのオブジェクトのソースコードをリバースエンジニアする（図3の成分31）。次いで、ソースコードをパースし、複素パラメータを宣言するストリングを抽出する（図3の成分32）。この目的のために、標準探索アルゴリズムまたは簡単な状態図に基づくパーザーを使用することができる。次に、パースされたストリングを変更して、パラメータ宣言ステートメントに対抗する変数宣言ステートメントにする。得られたストリングは、匿名ブロックの宣言区分を格納するストリングに付加される。

【0030】

複素型の宣言のためのストリングを構成した後に、本発明はサブプログラムの

パラメータを保持しているネストされたリストにアクセスし、複素型のサブコンポーネントを保持しているリストを入手する（図3の成分33）。次いで、複素パラメータの第1のスカラサブコンポーネントから始めて、リストを走査する。各スカラサブコンポーネント毎に、図1の成分12に入力された値を使用して初期化ステートメントが構成される。次いで、この初期化ステートメントが、匿名ブロックの初期化区分を保持しているストリングに付加される（図4の成分44）。

【0031】

スカラ型を有するパラメータの場合、本発明は図4の成分41においてそのパラメータの宣言を構成する。この目的のために、オラクルカタログ（図1の成分18）から検索したパラメータ情報が使用される。次いで本発明は、図4の成分44を使用して初期化ステートメントを構成する。

【0032】

匿名ブロックの宣言及び初期化区分が完了すると、本発明は、初期化区分と宣言区分とを組合せてPL／SQLコードの部分ブロックを保持するストリングを形成する。

【0033】

次いで、本発明は、図1の成分11からのサブプログラム名、及び図1の成分18からのパラメータ名を使用して、サブプログラムを実行するためのコードを構成する。次いで、このストリングを、図4の成分43において構成された部分匿名ブロックを含むストリングに付加する。

【0034】

次に、本発明は、出力パラメータの値を出力するための呼出しを構成する。この目的のために、パラメータのリストを走査するプロセスが再度適用される。各スカラパラメータ毎に、それが出力パラメータであるか否かが調べられ、もし出力パラメータであれば、値を出力するために呼出しに付加される。複素パラメータの場合には、サブコンポーネントを走査するプロセスが繰り返される。次いで、そのパラメータのサブコンポーネントの出力をフォーマットするために、単一の呼出しを構成する。

【0035】

以下は、格納された手順のための好ましい実施の形態内で生成される匿名ブロックの例である。

格納された手順 'foo' は、2つのパラメータを有している。第1のパラメータは、型VARCHAR2のINパラメータである。第2のパラメータは、型 'tab1 %TYPE' の複素パラメータである。但し、テーブル 'tab1' は2つの列、即ち型VARCHAR2(30)の第1の列 'col1'、及び型NUMBERの第2の列 'col2' を有している。以上の手順を適用することによって生成された匿名ブロックは以下のようになる。

```
1. declare ;
2. scalar1 varchar2(2000) ;
3. record tab1 %TYPE ;
4. begin ;
5. scalar1 := 'input 1' ;
6. record1.col1 := 'input 2' ;
7. record1.Col2 := input 3 ;
8. Foo(scalar1, record1) ;
9. dbms_output. line(record1.col1 || ',' || record1.Col2) ;
11. end ;
```

【0036】

上記第2行は、図1の成分18内で構成されたパラメータリストを使用することによって構成される。上記第3行は、図3の成分32におけるプロセスを使用して、ソースコードの宣言出力をパースすることによって構成される。第5行乃至第7行は、図4に示すプロセスを使用して構成される。第8行は、パラメータリスト内に格納されているパラメータ名を使用することによって構成され、OUTパラメータを出力する第9行は、パラメータリストを走査し、それらがOUTパラメータであるか否かを調べることによって構成される。他の関係型データベース管理システム(RDBMS)は、他の関係によって読出し専用、書込み専用、及び読取り／書込み制約と呼ぶことができる。

【0037】

オラクルデータベース、PL/SQL、及びSQL*Plusに関する付加的な背景情報については、以下を参照されたい。

1. SQL*Plus User's Guide and Reference Release 3.1 p 645.
2. PL/SQL User's Guide and Reference Release 2.2 Chapter 8.
3. Oracle 7 Application Developer's Guide Release 7.3.

【0038】

以上に、本発明を好ましい実施の形態に関して説明したが、当分野に精通していれば、同じ結果を達成するためには上述した構成に代えていろいろな型の汎用コンピュータハードウェアを使用できることが理解されよう。同様に、演算論理回路は、パラメトリック入力を処理して、格納されているデータオブジェクトを実行するために必要な呼出しを自動的に生成するための特許請求の範囲に記載の各必要手段を遂行するように構成されていることを理解されたい。当分野に精通していれば、特許請求の範囲から逸脱することなく、好ましい実施の形態の変更及び変化が可能であることは理解されよう。

【図面の簡単な説明】

【図1】

本発明の好ましい実施の形態の総合的な流れのフローチャートである。

【図2】

パラメータのリストの構造を示すフローチャートである。

【図3】

複素パラメータ宣言を構成するために使用されるステップを示すフローチャートである。

【図4】

好ましい実施の形態における宣言構成ステップを示すフローチャートである。

【図5】

好ましい実施の形態における匿名ブロックを構成するために使用されるステップのフローチャートである。

【図6】

本発明の好ましい実施の形態のクライアントPC及び／またはサーバーの何れかとして使用可能な代表的汎用コンピュータ構成を示す図である。

【図1】

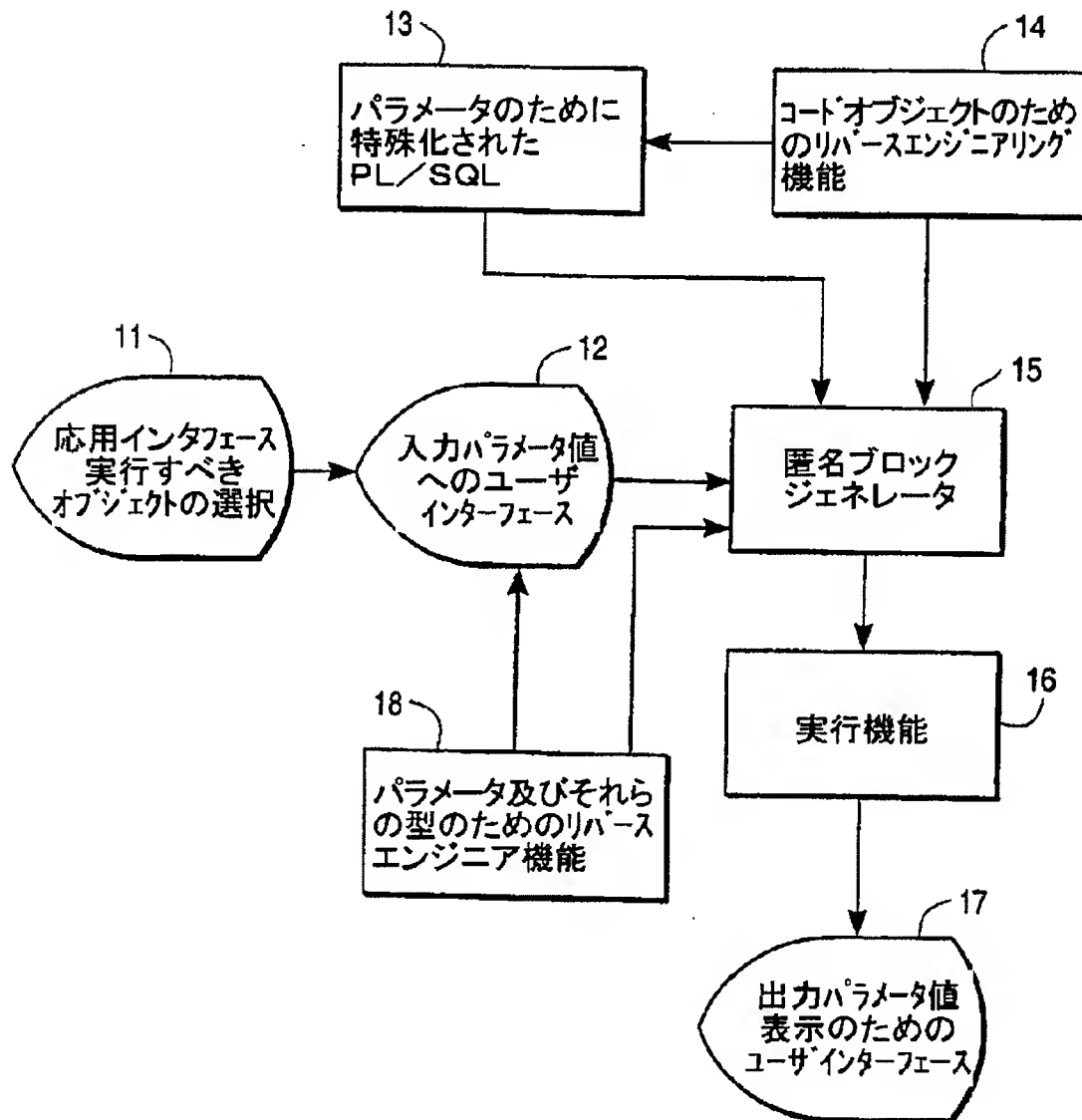


FIG. 1

【図2】

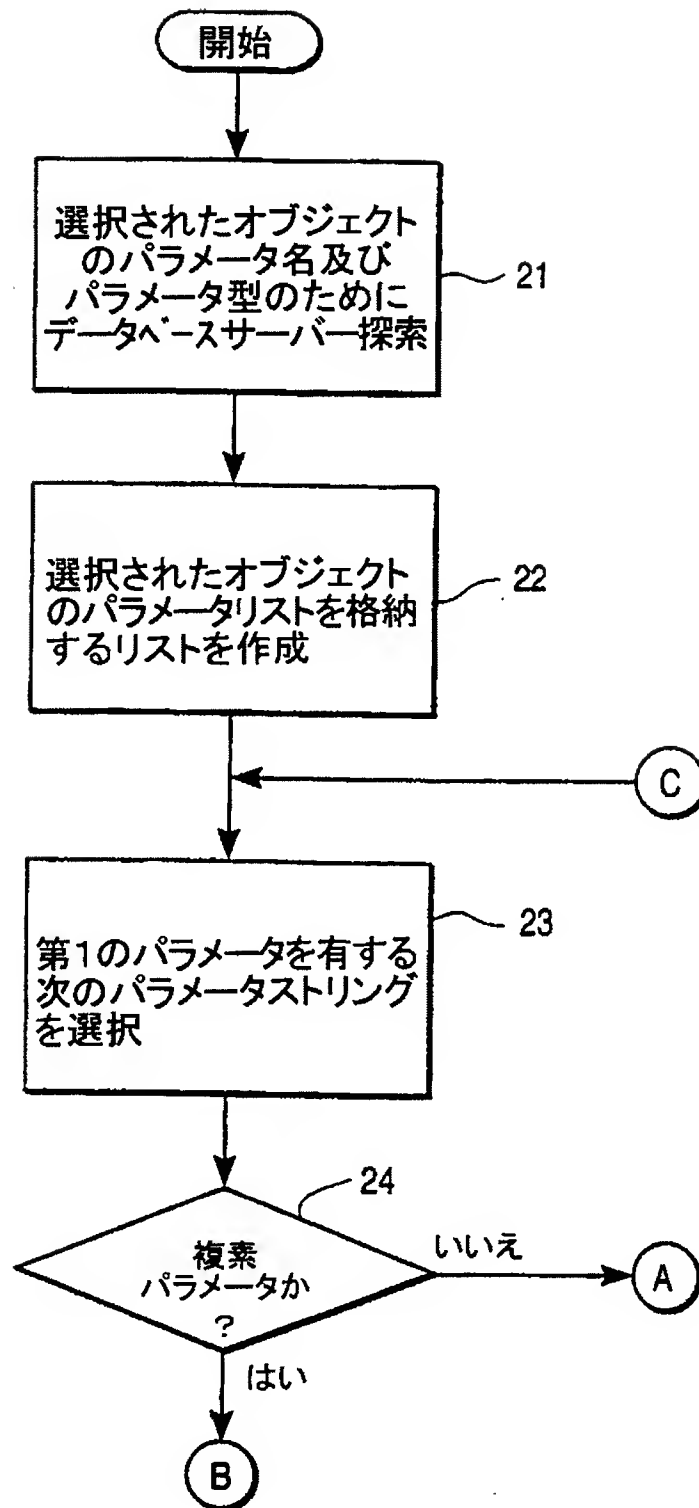
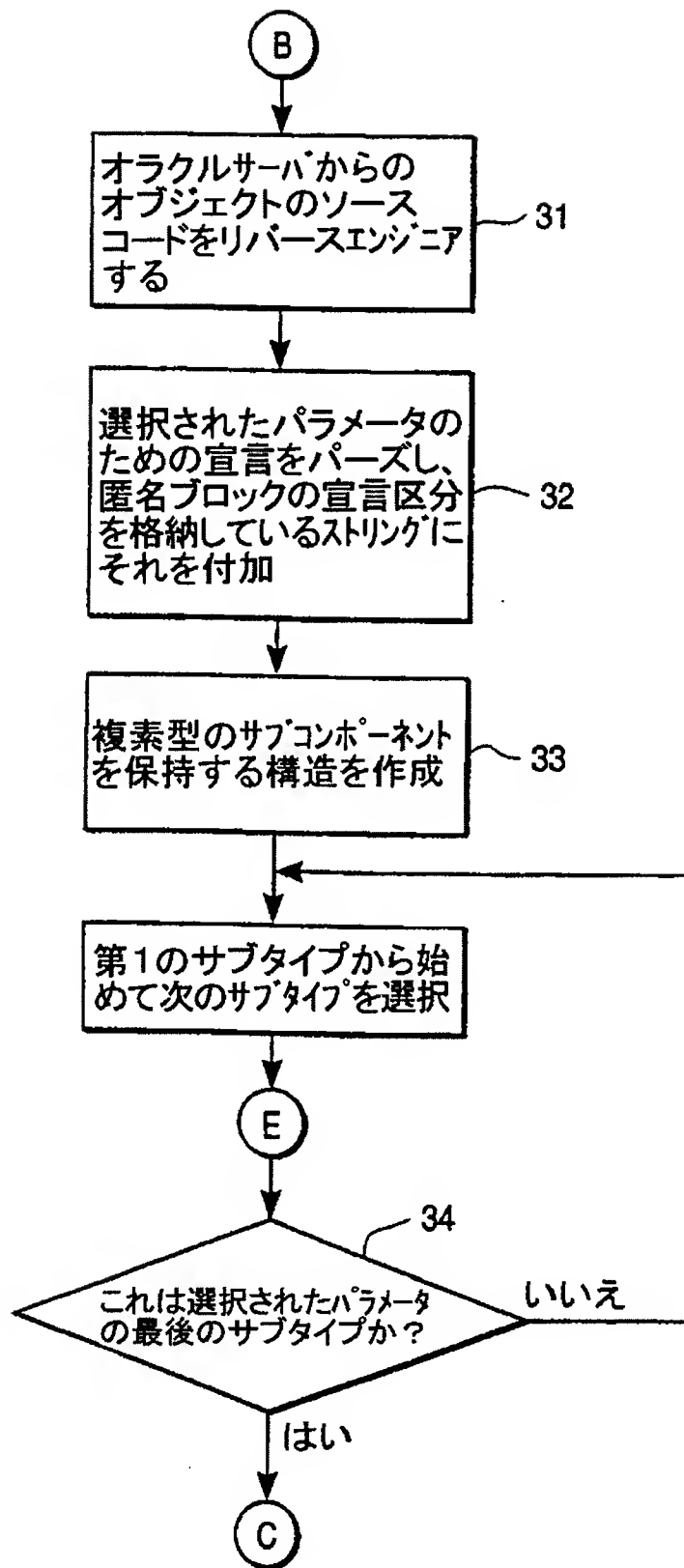


FIG. 2

【図3】



【図4】

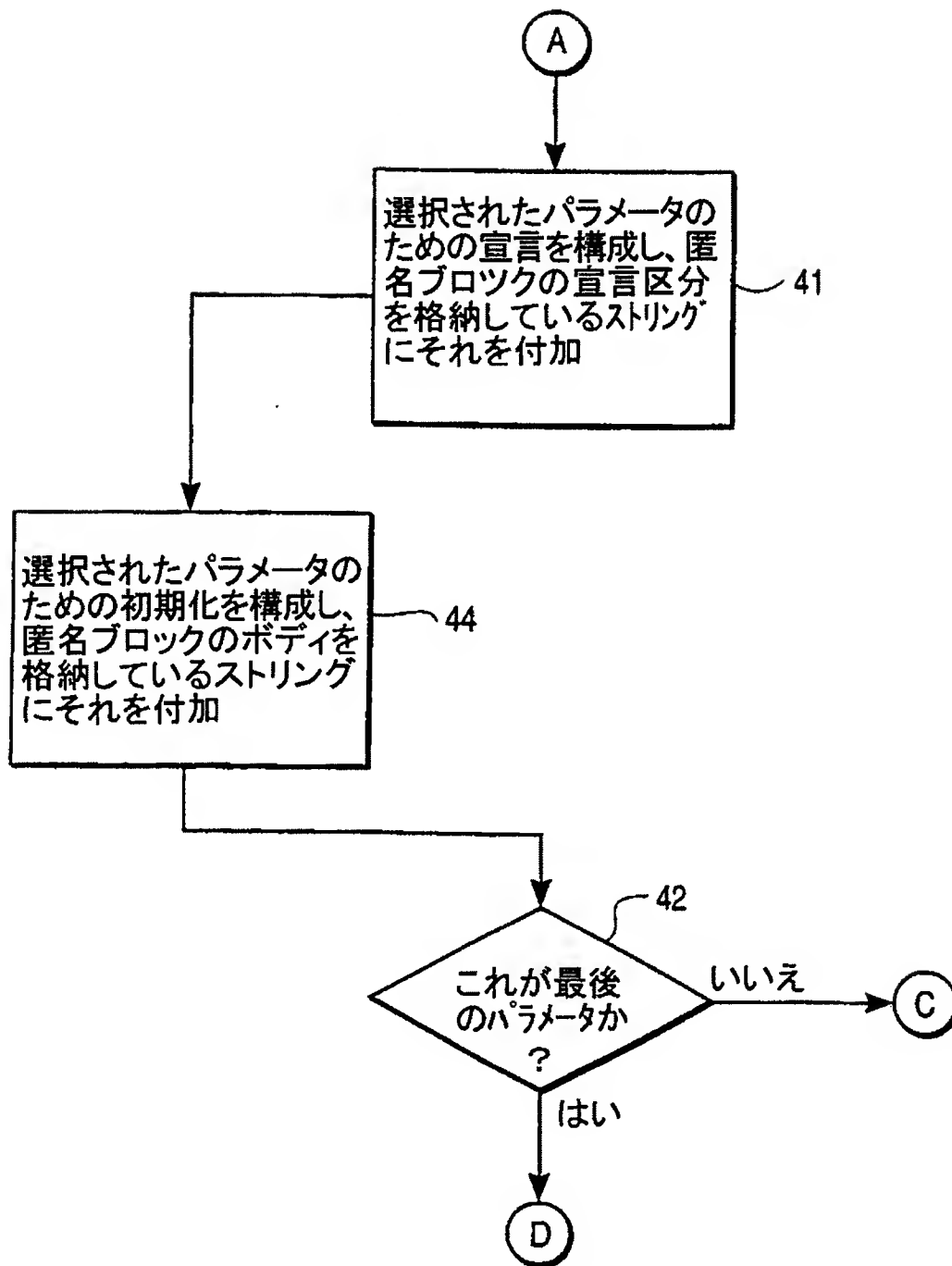


FIG. 4

【図5】

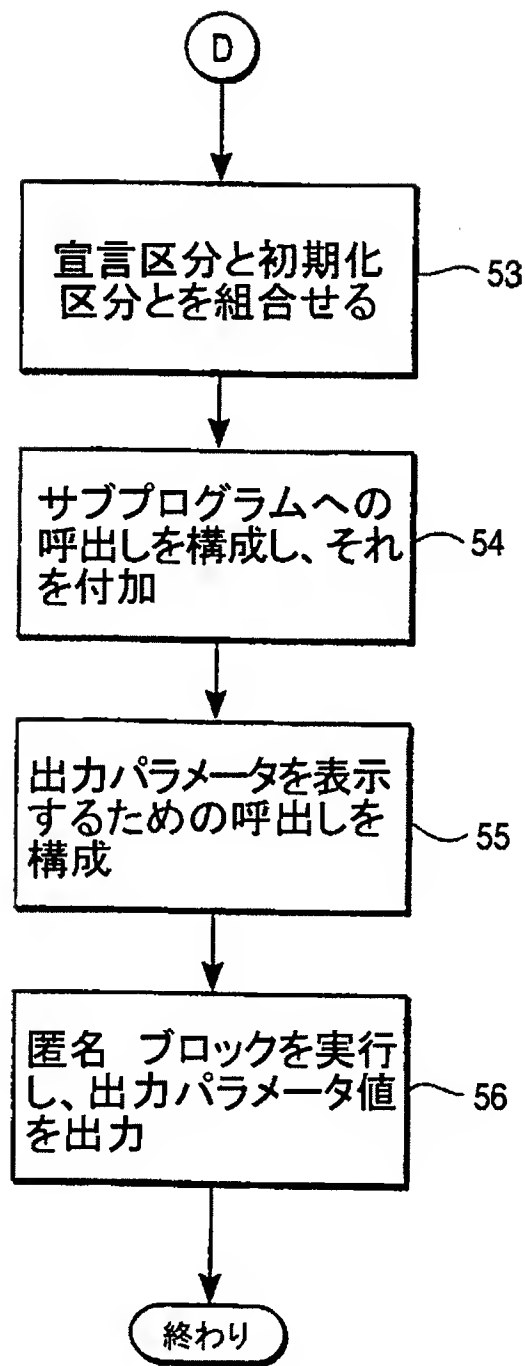


FIG. 5

【図6】

例示汎用コンピュータ 200

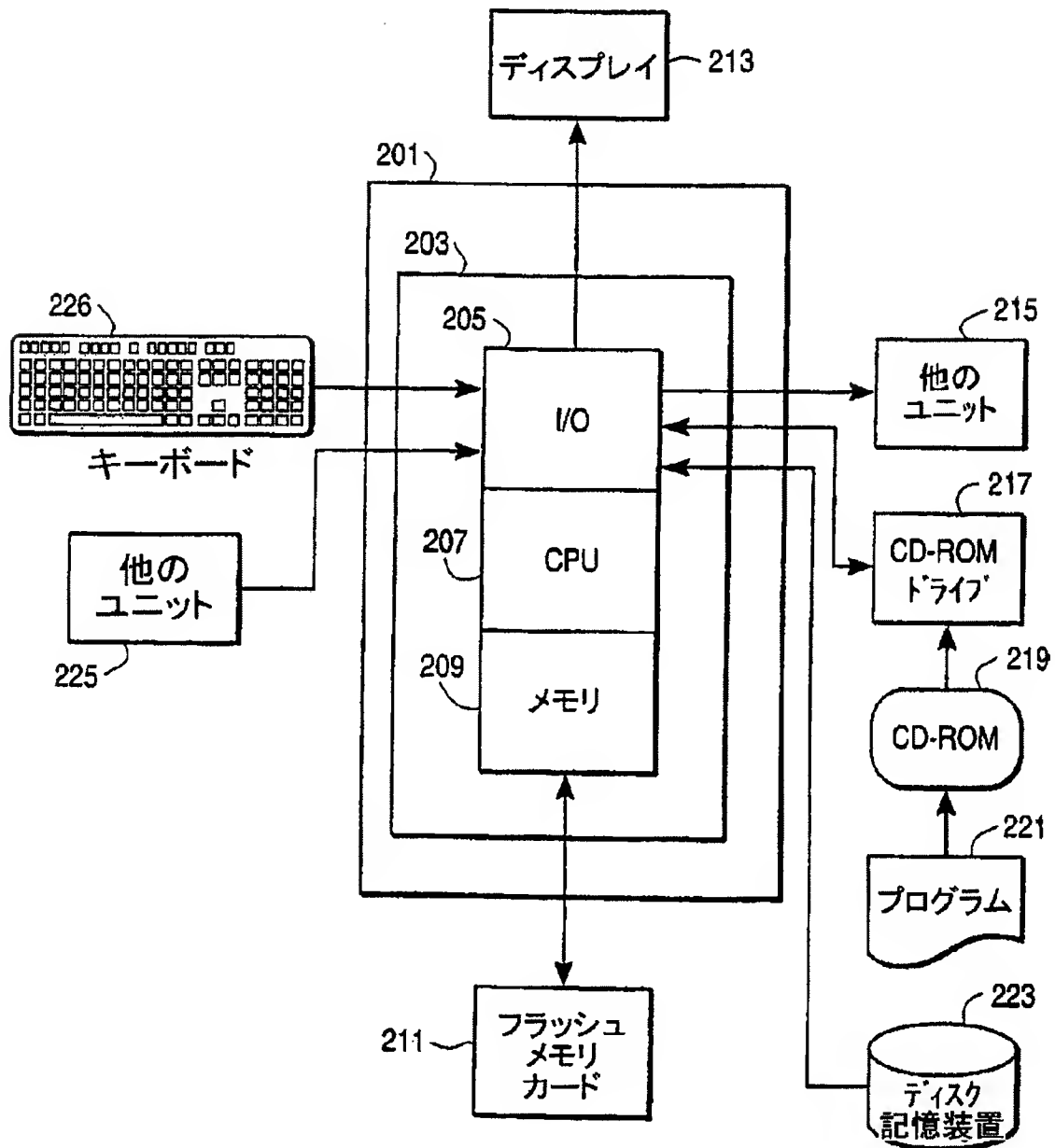


FIG. 6

【国際調査報告】

INTERNATIONAL SEARCH REPORT		International application No. PCT/US99/29921
A. CLASSIFICATION OF SUBJECT MATTER IPC(7) : G06F 9/45 US CL : 707/103 According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) U.S. : 707/103; 707/1-4, 200-204; 714/3; 395/701,703,704,708 Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Electronic data bases consulted during the international search (name of data base and, where practicable, search terms used) EAST, IEEE, NPL (ELSEVIER), ACM, DIALOGWEB		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 4,953,084 A (MELOY et al) 28 AUGUST 1990, col. 1, line 60 through col. 2, line 19; col. 9, lines 5-14	1-3, 9-10, 12
Y	US 5,361,351 A (LENKOV et al) 01 NOVEMBER 1994, Abstract, col. 1, lines 54-68, col. 9, lines 39-46, and col. 11, lines 20-25,	1-3, 9-10, 12
A	US 5,802,523 A (JASUJA et al) 01 SEPTEMBER 1998, the entire document	1-12
A,P	US 5,862,378 A (WANG et al) 19 JANUARY 1999, the entire document	1-12
A,P	US 6,003,039 A (BARRY et al) 14 DECEMBER 1999, the entire document	1-12
<input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "B" earlier document published on or after the international filing date "L" document which may throw doubt on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "Z" document member of the same patent family		
Date of the actual completion of the international search 20 MARCH 2000		Date of mailing of the international search report 18 APR 2000
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230		Authorized officer HOSAIN T. ALAM <i>Joni Hill</i> Telephone No. (703) 308-6662

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/29921

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim
Y	Paradigm Plus, Forward Engineering with Oracle8, Release 3.52, pages 1-38, published November, 1997	1-3, 9-10, 12
Y	Paradigm Plus, Round-trip Engineering with ERwin, Release 3.6, Platinum Technology, published August 1998	1-3, 9-10, 12

フロントページの続き

(81)指定国 EP(AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG), AP(GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW